UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/675,817	09/28/2000	Thomas Tomazin	10559-284001 / P9291-ADI	9781

20985 7590 07/08/2005

FISH & RICHARDSON, PC
12390 EL CAMINO REAL
SAN DIEGO, CA 92130-2081

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 07/08/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/675,817	TOMAZIN ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 April 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _____ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4-8,18-21 and 24-27* is/are rejected.

7)☒ Claim(s) *3 and 23* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 September 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *05 May 2005*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 3-8, 18-21, and 23-27 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Appeal Brief as received on 4/18/2005 and IDS as received on 5/5/2005.

Response to Arguments

3. In view of the appeal brief filed on April 18, 2005, PROSECUTION IS HEREBY REOPENED. A new grounds of rejection is set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) request reinstatement of the appeal.

If reinstatement of the appeal is requested, such request must be accompanied by a supplemental appeal brief, but no new amendments, affidavits (37 CFR 1.130, 1.131 or 1.132) or other evidence are permitted. See 37 CFR 1.193(b)(2).

Specification

4. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Drawings

5. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the concepts described in claims 3, 6, 18, 19, 23, and 26 must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

6. Claim 21 is objected to because of the following informalities: In line 4, replace "storing" with --store--. In line 10, replace "determining" with --determine--. In line 11, replace "decoding" with --decode--. In line 12, replace "determining" with --determine--. Appropriate correction is required.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 6, 18-20, 25-27 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

9. Claims 6 and 19 are not clear to the examiner because applicant does not state what is being aligned and what the aligning is done ahead of. If something is done ahead of something else, for the claim to be understood, the "something else" must be known. For instance, if the number of cycles in claim 6 (or claim 19) happened to be 3 cycles, the claim would essentially read "...aligning ahead 3 cycles." This meaning of this is not clear.

10. Claim 18 recites the limitation "the transition detector" in line 1. There is insufficient antecedent basis for this limitation in the claim.

11. Claim 25 is not clear because:

a) in lines 7-8, it is stated that at least one of said cache subportions is/are selected as a

current instruction. However, a subportion cannot be a software instruction since it is a portion of a hardware buffer. Therefore, applicant should amend the claim such that the contents of the subportion(s) are selected as a current instruction or that the subportion(s) is/are selected to provide a current instruction.

b) in the last 3 lines of claim 25, it is not clear what "loading of that number of buffer areas" refers to because applicant previously mentions "a number of cycles". So, if there are 3 cycles, does the system load 3 buffers? Instead of claiming "loading of that number of buffer areas" applicant should refer to loading of the buffer areas that are predicted to be depleted. If applicant feels that such a change is in appropriate, the examiner requests some other clarifying amendment.

12. Claim 27 is not clear because in lines 7-8, it is stated that at least one of said subparts is aligned as a current instruction. However, a subpart cannot be a software instruction since it is a portion of a hardware buffer. Therefore, applicant should amend the claim such that the contents of the subpart(s) are aligned as a current instruction (other language to this effect is sufficient).

Claim Rejections - 35 USC § 102

13. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

14. Claims 1, 4-6, 8, 19, 21, 24-25, and 27 are rejected under 35 U.S.C. 102(b) as being anticipated by Martin, U.S. Patent No. 4,439,828.

15. Referring to claims 1, Martin has taught a method of aligning instructions in a processor comprising:

a) storing a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more that sub buffer. See Fig.2 and note buffer areas 18, 19, and 20. Each buffer area includes a plurality of sub-buffers 21, which in turn store portions of instructions having different sizes. See column 3, lines 61-67.

b) aligning a first instruction from said buffer areas. See Fig.2, and note that no matter buffer area(s) the first instruction is stored in, the instruction will be selected, and sent to the instruction register where it will be decoded. See column 4, lines 49-54.

c) decoding the size of the first instruction. See column 4, lines 12-17. Note that the size of the first instruction is determined (decoded) and used by the instruction counter 22.

d) selecting at least one of said plurality of sub-buffers from which to output said first instruction on an output part. See Fig.2, and note that the sub-buffers holding the desired instruction will output the instruction portions on the respective wires, which connect the sub-buffers to the instruction register 24.

e) during said outputting, determining a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction, decoding the size of the second instruction, and determining whether processing the second instruction will deplete said plurality of buffer areas. See column 4, lines 12-23. Note that the size of the first instruction is used to increment the instruction counter 22 so that it points to the second instruction. Likewise, when the second instruction is retrieved, its size will be determined so that the instruction

counter may locate the third instruction, etc. In addition, it is determined whether the processing of instructions will empty/deplete the buffer.

f) based on said determining whether processing the second instruction will deplete said plurality of buffer areas, instructing the plurality of buffer areas to receive additional instructions. See column 4, lines 17-23, and note that buffer depletion results in replenishing the buffer areas with instructions from main memory.

16. Referring to claim 4, Martin has taught a method as described in claim 1. Martin has further taught storing a first instruction across a plurality of sub-buffers prior to processing the instructions. See Fig.2 and column 3, lines 61-67.

17. Referring to claim 5, Martin has taught a method as described in claim 1. Martin has further taught adding the size of the first instruction to a current instruction position to determine the beginning of the second instruction. See column 4, lines 12-17.

18. Referring to claim 6, Martin has taught a method as described in claim 1. Martin has further taught aligning ahead a number of cycles equal to a cache latency. Assuming that a new instruction is aligned every clock cycle (column 1, lines 15-18), then instructions will be aligned at times N, N+1, N+2,..., N+M-1, and N+M. In this situation, the cache latency is 1 and an alignment occurs 1 cycle ahead. Fore instance, a first instruction is aligned at time N, which is 1 cycle ahead of N+1 (current time + cache latency).

19. Referring to claim 8, Martin has taught a method as described in claim 1. Martin has further taught issuing a request to a memory to reload the plurality of buffer areas. See column 4, lines 20-23.

20. Referring to claim 19, Martin has taught a processor as described in claim 27. Martin has further taught aligning ahead a number of cycles equal to a cache latency. Assuming that a new instruction is aligned every clock cycle (column 1, lines 15-18), then instructions will be aligned at times N, N+1, N+2,..., N+M-1, and N+M. In this situation, the cache latency is 1 and an alignment occurs 1 cycle ahead. Fore instance, a first instruction is aligned at time N, which is 1 cycle ahead of N+1 (current time + cache latency).

21. Referring to claim 21, Martin has taught an apparatus, including instructions residing on a machine-readable storage medium, for use in a machine system to align instructions in a processor, the instructions causing the machine to:

a) store a plurality of instruction of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer. See Fig.2 and note buffer areas 18, 19, and 20. Each buffer area includes a plurality of sub-buffers 21, which in turn store portions of instructions having different sizes. See column 3, lines 61-67.

b) decode a size of a first instruction from said plurality of buffer areas. See column 4, lines 12-17. Note that the size of the first instruction is determined (decoded) and used by the instruction counter 22.

c) select at least one of said plurality of sub-buffers from which to output said first instruction on an output part. See Fig.2, and note that the sub-buffers holding the desired instruction will output the instruction portions on the respective wires, which connect the sub-buffers to the instruction register 24.

d) during said outputting, determine a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction, decode the size of the second instruction, and determine whether processing the second instruction will deplete said plurality of buffer areas. See column 4, lines 12-23. Note that the size of the first instruction is used to increment the instruction counter 22 so that it points to the second instruction. Likewise, when the second instruction is retrieved, its size will be determined so that the instruction counter may locate the third instruction, etc. In addition, it is determined whether the processing of instructions will empty/deplete the buffer.

e) based on said determining whether processing the second instruction will deplete said plurality of buffer areas, instruct the plurality of buffer areas to receive additional instructions. See column 4, lines 17-23, and note that buffer depletion results in replenishing the buffer areas with instructions from main memory.

22. Referring to claim 24, Martin has taught an apparatus as described in claim 21. Martin has further taught storing a first instruction across a plurality of sub-buffers prior to processing the instructions. See Fig.2 and column 3, lines 61-67.

23. Referring to claim 25, Martin has taught a method of processing instructions within a processor, comprising:

a) storing instructions of different widths within a cache having a plurality of buffer areas, each buffer area having a plurality of subportions, each subportion in the cache storing a unit instruction width, where an instruction of unit width takes up a single subportion in the cache, and an instruction of more than said unit width takes up more than one subportion within the cache. See Fig.2 and note buffer areas 18, 19, and 20. Each buffer area includes a plurality of

subportions 21, which in turn store portions of instructions having different widths. See column 3, lines 61-67.

b) multiplexing each of the subportions of said cache to an output point, and selecting at least one of said cache subportions as a current instruction. See Fig.2 and column 4, lines 31-37. Note that instructions are sent to gates 29 and selector 26 chooses which gates to enable such that the associated instructions are outputted. Gates 29 and selector 26 perform multiplexing because multiplexing is the selection of an item from a plurality of items and any of the instructions in the plurality of subportions may be selected.

c) during said selecting said current instruction, predicting which of said buffer areas within said cache will be depleted of instruction data within a number of cycles approximately equal to a latency of the cache, and instructing loading of that number of buffer areas with additional instruction information. See column 4, lines 12-23, and note that it is determined whether the processing of instructions will empty/deplete the buffer. Note further that buffer depletion results in replenishing the buffer areas with instructions from main memory. The system, upon selecting an instruction each cycle, determines that a portion holding the selected instruction will emptied, and therefore should be refilled.

24. Referring to claim 27, Martin has taught a processor comprising:

a) a plurality of buffer areas, each buffer area adapted to store a plurality of instructions of different widths in a plurality of subparts, each of said subparts storing a unit instruction width, and said instruction of greater than a unit instruction width being stored in multiple said subparts. See Fig.2 and note buffer areas 18, 19, and 20. Each buffer area includes a plurality of subparts

21, which in turn store portions of instructions having different widths. See column 3, lines 61-67.

b) a multiplexer, connected to said plurality of subparts, and selecting and aligning at least one of said subparts from any of said subparts within said buffer areas as a current instruction. See Fig.2 and column 4, lines 31-37. Note that instructions are sent to gates 29 and selector 26 chooses which gates to enable such that the associated instructions are outputted. Gates 29 and selector 26 are a multiplexer because a multiplexer selects an item from a plurality of items, and Martin's components perform such a task in that any of the items in the plurality of subparts may be selected for processing.

c) a predictor, operating to predict when at least one of the plurality of buffer areas will be empty, and to send a signal to instruction said at least one of the plurality of buffer areas to load another instruction data. See column 4, lines 12-23, and note that it is determined whether the processing of instructions will empty/deplete the buffer. Note further that buffer depletion results in replenishing the buffer areas with instructions from main memory. The system, upon selecting an instruction each cycle, determines that a portion holding the selected instruction will emptied, and therefore should be refilled.

Claim Rejections - 35 USC § 103

25. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

26. Claims 1, 4-6, 8, 21, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zuraski, Jr. et al., U.S. Patent No. 6,260,134 (as applied in the previous Office Action and herein referred to as Zuraski) in view of Martin, as applied above.

27. Referring to claim 1, Zuraski has taught a method of aligning instructions in a processor comprising:

a) storing a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more that sub buffer. See Fig.4, component 302, and note that instructions are buffered. Also, looking at Fig.1B, it can be seen that different length instructions are present. The instruction buffer inherently has many bits of storage. Each bit is a sub-buffer which holds 1-bit of an instruction (unit instruction width). A group of these bits makes up a buffer area. And, clearly, there are multiple buffer areas. For instance, each buffer area may be 8 bits, where each buffer area includes eight 1-bit sub-buffers.

b) aligning a first instruction from said buffer areas. See Fig.2, component 18, and Fig.4, and column 6, lines 20-29.

c) decoding the size of the first instruction. See the abstract, Fig.4, components 306, 308, and 310, and column 12, line 63, to column 13, line 17.

d) selecting at least one of said plurality of sub-buffers from which to output said first instruction on an output part. See Fig.4 and note that the instruction buffer outputs instructions through the multiplexer 304 to other additional components (306, 308, etc.). Again, the sub-buffers hold instructions to they will be selected to output the instructions.

e) during said outputting, determining a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction. See Fig.4, component 312, and column 13, lines 18-22

f) decoding the size of the second instruction. See the abstract, Fig.4, components 306, 308, and 310, and column 12, line 63, to column 13, line 17. Note that the same process is repeated for each instruction.

g) Zuraski has not taught determining whether processing the second instruction will deplete said plurality of buffer areas based on said determining whether processing the second instruction will deplete said plurality of buffer areas, instructing the plurality of buffer areas to receive additional instructions. However, Martin has taught the concept of refilling a buffer in response to the emptying of the buffer due to instruction processing. See column 4, lines 17-23, and note that buffer depletion results in replenishing the buffer areas with instructions from main memory. Prefetching increases the speed of execution by keeping the execution units busy and not stalled or waiting on instructions from memory, which operates at a slower pace. Therefore, by fetching instruction ahead of time, before they are needed and always making instructions available to execution units, the execution units, or pipeline, will not stall, thus decreasing the amount of time needed for execution. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to use prefetching to decrease the amount of time needed for execution, and to keep the prefetch buffer full when it is depleted.

28. Referring to claim 4, Zuraski in view of Martin has taught a method as described in claim 1. Zuraski has further taught storing a first instruction across a plurality of sub-buffers prior to

processing the instructions. Fig.1B shows that instructions are at least 8 bits wide. If each sub-buffer is 1-bit of storage, then an instruction would require multiple sub-buffers.

29. Referring to claim 5, Zuraski in view of Martin has taught a method as described in claim 1. Zuraski has further taught adding the size of the first instruction to a current instruction position to determine the beginning of the second instruction. See column 13, lines 18-22.

30. Referring to claim 6, Zuraski in view of Martin has taught a method as described in claim 1. Zuraski in view of Martin has further taught aligning ahead a number of cycles equal to a cache latency. Since Martin fills the buffer so that it will never be empty, the fetching is done equal to a cache latency, therefore allowing the alignment to occur at a cache latency since the alignment of Zuraski occurs after the instructions are fetched from memory.

31. Referring to claim 8, Zuraski in view of Martin has taught a method as described in claim 1. Martin has further taught issuing a request to a memory to reload the plurality of buffer areas. See column 4, lines 20-23.

32. Referring to claim 21, Zuraski has taught an apparatus, including instructions residing on a machine-readable storage medium, for use in a machine system to align instructions in a processor, the instructions causing the machine to:

a) store a plurality of instruction of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer. See Fig.4, component 302, and note that instructions are buffered. Also, looking at Fig.1B, it can be seen that different length instructions are present. The instruction buffer inherently has many bits of storage. Each bit is a sub-buffer which holds 1-bit of an instruction (unit instruction width). A

group of these bits makes up a buffer area. And, clearly, there are multiple buffer areas. For instance, each buffer area may be 8 bits, where each buffer area includes eight 1-bit sub-buffers.

b) decode a size of a first instruction from said plurality of buffer areas. See the abstract, Fig.4, components 306, 308, and 310, and column 12, line 63, to column 13, line 17.

c) select at least one of said plurality of sub-buffers from which to output said first instruction on an output part. See Fig.4 and note that the instruction buffer outputs instructions through the multiplexer 304 to other additional components (306, 308, etc.). Again, the sub-buffers hold instructions to they will be selected to output the instructions.

d) during said outputting, determine a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction. See Fig.4, component 312, and column 13, lines 18-22

e) decode the size of the second instruction. See the abstract, Fig.4, components 306, 308, and 310, and column 12, line 63, to column 13, line 17. Note that the same process is repeated for each instruction.

f) Zuraski has not taught determining whether processing the second instruction will deplete said plurality of buffer areas and based on said determining whether processing the second instruction will deplete said plurality of buffer areas, instruct the plurality of buffer areas to receive additional instructions. However, Martin has taught the concept of refilling a buffer in response to the emptying of the buffer due to instruction processing. See column 4, lines 17-23, and note that buffer depletion results in replenishing the buffer areas with instructions from main memory. Prefetching increases the speed of execution by keeping the execution units busy and not stalled or waiting on instructions from memory, which operates at a slower pace. Therefore, by fetching

instruction ahead of time, before they are needed and always making instructions available to execution units, the execution units, or pipeline, will not stall, thus decreasing the amount of time needed for execution. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to use prefetching to decrease the amount of time needed for execution, and to keep the prefetch buffer full when it is depleted.

33. Referring to claim 24, Zuraski in view of Martin has taught an apparatus as described in claim 21. Zuraski has further taught storing a first instruction across a plurality of sub-buffers prior to processing the instructions. Fig.1B shows that instructions are at least 8 bits wide. If each sub-buffer is 1-bit of storage, then an instruction would require multiple sub-buffers.

34. Claims 7 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Martin, as applied above, in view of Davis, U.S. Patent Number 6,367,003 (as applied in the previous Office Action and herein referred to as Davis).

35. Referring to claim 7, Martin has taught a method as described in claim 1. Martin has not explicitly taught aligning instructions in a digital signal processor. However, Davis has taught providing instructions in a digital signal processor (DSP). See column 1, lines 21-36. A DSP specializes in executing specific types of algorithms typically encountered in signal processing such as multiply-accumulate operations. Therefore, in order to optimize execution of digital signal applications in Martin, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Martin to be a digital signal processor.

36. Referring to claim 20, Martin has taught a processor as described in claim 27. Martin has not explicitly taught that the processor is a digital signal processor. However, Davis has taught

providing that a digital signal processor (DSP) specializes in executing specific types of algorithms typically encountered in signal processing such as multiply-accumulate operations. See column 1, lines 21-36. Therefore, in order to optimize execution of digital signal applications in Martin, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Martin to be a digital signal processor.

37. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Zuraski in view of Martin, as applied above, in view of Davis, as applied above.

38. Referring to claim 7, Zuraski in view of Martin has taught a method as described in claim 1. Zuraski in view of Martin has not explicitly taught aligning instructions in a digital signal processor. However, Davis has taught providing instructions in a digital signal processor (DSP). See column 1, lines 21-36. A DSP specializes in executing specific types of algorithms typically encountered in signal processing such as multiply-accumulate operations. Therefore, in order to optimize execution of digital signal applications in Zuraski, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Zuraski to be a digital signal processor.

39. Claims 19, 25, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan et al., U.S. Patent No. 6,161,172 (as applied in the previous Office Action and herein referred to as Narayan) in view of Nishii et al., U.S. Patent No. 5,918,045 (as applied in the previous Office Action and herein referred to as Nishii).

40. Referring to claim 19, Narayan in view of Nishii has taught a processor as described in claim 27. Narayan in view of Nishii have further taught aligning ahead a number of cycles equal to a cache latency (Nishii column 8 lines 38-60). Since Nishii fills the prefetch buffer so that it will never be empty, the fetching is done equal to a cache latency, therefore allowing the alignment to occur at a cache latency since the alignment of Narayan occurs after the instructions are fetched from memory.

41. Referring to claim 25, Narayan has taught a method of processing instructions within a processor, comprising:

a) storing instructions of different widths within a cache having a plurality of buffer areas, each buffer area having a plurality of subportions, each subportion in the cache storing a unit instruction width, where an instruction of unit width takes up a single subportion in the cache, and an instruction of more than said unit width takes up more than one subportion within the cache. See Fig.4, components 86A-C (buffer areas), column 6, lines 23-65, and column 17, line 61, to column 18, line 4. Clearly, if each buffer area holds bytes of instructions, then each buffer area has a plurality of byte subportions (it may also be interpreted that the buffer areas have many 1- bit subportions, 2-bit subportions, etc., as applicant has not defined the subportion size). It should also be noted from column 18, lines 34-41, that instructions are variable length. So, if the subportions are interpreted as being able to store a byte of information, then an X-byte instruction will require X subportions of storage whereas a Y-byte instruction will require Y subportions of storage.

b) multiplexing each of the subportions of said cache to an output point, and selecting at least one of said cache subportions as a current instruction. See Fig.4 and Fig.6, component 78, column 20, lines 18-33, and column 23, lines 3-27.

c) Narayan has not taught that during said selecting said current instruction, predicting which of said buffer areas within said cache will be depleted of instruction data within a number of cycles approximately equal to a latency of the cache, and instructing loading of that number of buffer areas with additional instruction information. However, Nishii has taught such a concept. See column 7, line 61, to column 8, line 4, and column 8, lines 38-60, and note that in Nishii, the prefetch buffer is always kept from being empty by comparing read/write two pointers and determining when the buffer needs to have more instructions fetched from memory. The system in order to keep the buffer full at all times, must know when the next instruction, or number of instructions, will deplete the buffer. Prefetching increases the speed of execution by keeping the execution units busy and not stalled or waiting on instructions from memory, which operates at a slower pace (Nishii column 1, lines 5-10). Therefore, by fetching instruction ahead of time, before they are needed and always making instructions available to execution units, the execution units, or pipeline, will not stall, thus decreasing the amount of time needed for execution. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to use prefetching to decrease the amount of time needed for execution, and to keep the prefetch buffer full when it is depleted.

42. Referring to claim 27, Narayan has taught a processor comprising:

a) a plurality of buffer areas, each buffer area adapted to store a plurality of instructions of different widths in a plurality of subparts, each of said subparts storing a unit instruction width,

and said instruction of greater than a unit instruction width being stored in multiple said subparts. See Fig.4, components 86A-C (buffer areas), column 6, lines 23-65, and column 17, line 61, to column 18, line 4. Clearly, if each buffer area holds bytes of instructions, then each buffer area has a plurality of byte subparts (it may also be interpreted that the buffer areas have many 1- bit subparts, 2-bit subparts, etc., as applicant has not defined the subpart size). It should also be noted from column 18, lines 34-41, that instructions are variable length. So, if the subparts are interpreted as being able to store a byte of information, then an X-byte instruction will require X subparts of storage whereas a Y-byte instruction will require Y subparts of storage.

b) a multiplexer, connected to said plurality of subparts, and selecting and aligning at least one of said subparts from any of said subparts within said buffer areas as a current instruction. See Fig.4 and Fig.6, component 78, column 20, lines 18-33, and column 23, lines 3-27.

c) Narayan has not taught a predictor, operating to predict when at least one of the plurality of buffer areas will be empty, and to send a signal to instruction said at least one of the plurality of buffer areas to load another instruction data. However, Nishii has taught such a concept. See column 7, line 61, to column 8, line 4, and column 8, lines 38-60, and note that in Nishii, the prefetch buffer is always kept from being empty by comparing read/write two pointers and determining when the buffer needs to have more instructions fetched from memory. The system in order to keep the buffer full at all times, must know when the next instruction, or number of instructions, will deplete the buffer. Prefetching increases the speed of execution by keeping the execution units busy and not stalled or waiting on instructions from memory, which operates at a slower pace (Nishii column 1, lines 5-10). Therefore, by fetching instruction ahead of time, before they are needed and always making instructions available to execution units, the

execution units, or pipeline, will not stall, thus decreasing the amount of time needed for execution. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to use prefetching to decrease the amount of time needed for execution, and to keep the prefetch buffer full when it is depleted.

43. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan in view of Nishii, as applied above, in view of Davis, as applied above.

44. Referring to claim 20, Narayan in view of Nishii has taught a processor as described in claim 27. Narayan in view of Nishii has not explicitly taught that the processor is a digital signal processor. However, Davis has taught providing that a digital signal processor (DSP) specializes in executing specific types of algorithms typically encountered in signal processing such as multiply-accumulate operations. See column 1, lines 21-36. Therefore, in order to optimize execution of digital signal applications in Narayan in view of Nishii, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Narayan in view of Nishii to be a digital signal processor.

Allowable Subject Matter

45. Claims 3 and 23 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

46. Claims 18 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and

any intervening claims **and the appropriate corrections are made to overcome the 112 rejection(s) set forth for claims 18 and 27**.

47. Claims 26 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims **and the appropriate corrections are made to overcome the 112 rejection(s) set forth for claim 25**.

Response to Arguments

48. In the appeal brief filed on April 18, 2005, applicant argues that the prior art of record, in general does not teach a plurality of buffer areas, each buffer area having a plurality of sub-buffers, each sub-buffer storing a unit instruction width, where an instruction of greater than a unit instruction width is stored in multiple sub-buffers.

49. However, the examiner asserts that all of Narayan, Martin, and Zuraski teach this. While Martin teaches this idea explicitly as described in the rejection above, this concept is also inherent within Narayan and Zuraski. More specifically, Narayan teaches buffer areas 86 (Fig.4) for holding instructions. A buffer which holds instructions inherently has sub-portions, the size of which was not defined in the claims by applicant. Therefore, a sub-portion may be defined as a 1-bit storage. If this is the case, then each buffer area would have many subportions so that each buffer area may store many bits of information (which is required if the buffer areas are to hold instructions). If an instruction is 16 bits in length, then it will require 16 subportions. At the same time, a subportion could be considered as being 2 bits of storage, in which case, a 16-bit instruction would require 8 subportions, and so on. This reasoning also applies to the

instruction buffer 302 (Fig.4) of Zuraski. Zuraski's buffer also holds instructions and therefore has many 1-bit storage subportions. Any combination of these subportions may be referred to as buffer areas.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
June 30, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100